

WHAT IS CLAIMED IS:

1           1. A computer-implemented method of determining if a query element is  
2 included in a set of elements, the method comprising:

3                 building a data structure based upon information identifying elements in the  
4 set of elements;

5                 receiving information identifying the query element; and

6                 using the data structure to determine if the query element is included in the set  
7 of elements such that the number of comparisons needed to determine if the query element is  
8 included in the set of elements is proportional to a length of the query element and  
9 independent of the number of elements in the set of elements.

1           2. The method of claim 1 wherein the query element is of length “q” and  
2 at most “q” character comparisons are needed to determine if the query element is included in  
3 the set of elements.

1           3. The method of claim 1 wherein:

2                 the set of elements contains elements from a domain  $\Sigma$  having a character set  
3 of “m” characters, wherein “Z” is the maximum possible length of an element in domain  $\Sigma$   
4 and “Y” is the length of the longest element in the set of elements such that  $1 \leq Y \leq Z$ ; and

5                 building the data structure comprises building the data structure comprising a  
6 plurality of memory structures headed by a root memory structure, each memory structure in  
7 the plurality of memory structures comprising a first memory location and an array of “m”  
8 memory locations.

1           4. The method of claim 3 wherein:

2                 the data structure comprises a total of  $(Y+1)$  levels; and

3                 each memory structure in the data structure belongs to a level L, where  $(0 \leq L$   
4  $\leq Y)$ , the level for a particular memory structure denoting the number of memory structures,  
5 starting with the root memory structure, that have to be traversed to reach the particular  
6 memory structure, the root memory structure belonging to level 0.

1           5. The method of claim 4 wherein building the data structure based upon  
2 information identifying the elements in the set of elements comprises:

3                   for each element “R” in the set of elements, where  $R = c_1c_2\dots c_f$  for some  
4                    $f \leq Y$ , for each  $c_i$  where  $1 \leq i \leq f$ , starting with  $i = 1$ :  
5                         (a) selecting a memory structure at level “(i-1)”;  
6                         (b) if a memory location corresponding to character  $c_i$  in the array of  
7                         memory locations of the presently selected memory structure does not refer to another  
8                         memory structure in the database, storing an address of a new memory structure at level “i”  
9                         in the memory location corresponding to character  $c_i$  in the array of memory locations of the  
10                        selected memory structure;  
11                         (c) selecting the memory structure at level “i” whose address is stored  
12                         in the memory location corresponding to character  $c_i$  in the array of memory locations of the  
13                         presently selected memory structure;  
14                         (d) if (“i” is equal to “f”), storing a reference to element “R” in the first  
15                         memory location of the memory structure selected in step (c);  
16                         (e) incrementing the value of “i” by one; and  
17                         (f) repeating steps (b), (c), (d), and (e) for each  $c_i$  where (“i”  $\leq$  “f”).

1                 6.     The method of claim 5 wherein:  
2                         receiving information identifying the query element comprises:  
3                         receiving information identifying a query element  $k$ , where  $k = c_1c_2\dots c_q$   
4                         for some  $q \leq Z$ ;  
5                         using the data structure to determine if the query element is included in the set  
6                         of elements comprises:  
7                         for each  $c_i$  of  $k$  where  $1 \leq i \leq f$ , starting with  $i = 1$ :  
8                         (a) selecting a memory structure of the database at level “(i-1)”;  
9                         (b) if a memory location corresponding to character  $c_i$  in the array of  
10                         memory locations of the presently selected memory structure does not refer to another  
11                         memory structure in the database, outputting a signal indicating that the query element is not  
12                         included in the set of elements;  
13                         (c) if the memory location corresponding to character  $c_i$  in the array of  
14                         memory locations of the presently selected memory structure stores an address of a memory  
15                         structure of the database at level “i”, selecting the memory structure at level “i” whose  
16                         address is stored;  
17                         (d) incrementing the value of “i” by one; and

7. The method of claim 3 wherein building the data structure based upon information identifying the elements in the set of elements comprises:

for each element “R” in the set of elements, where  $R = c_1c_2\dots c_f$  for some  
each character  $c_i$  belongs to the character set of domain  $\Sigma$ , and  $1 \leq i \leq f$ , storing  
in the database indicating the position and identity of each character in element

8. The method of claim 7 wherein using the data structure to determine if the query element is included in the set of elements comprises:

determining if the query element is included in the set of elements based upon information stored by the database and information identifying characters and their positions in the query element.

9. The method of claim 3 wherein building the data structure based upon information identifying the elements in the set of elements comprises:

for each element “R” in the set of elements:

(a) selecting the root memory structure of the data structure as the selected memory structure;

(b) selecting the first character of element R;

(c) if a memory location corresponding to the selected character in the string of the selected memory structure does not refer to another memory structure, storing an address of a new memory structure in the memory

10. The method of claim 9 wherein using the data structure to determine if the query element is included in the set of elements comprises:

- (a) selecting the root memory structure of the data structure as the selected memory structure;
- (b) selecting the first character of the query element;
- (c) if a memory location corresponding to the selected character in the array of memory locations of the selected memory structure does not refer to another memory structure in the data structure, outputting a signal indicating that the query element is not included in the set of elements,
  - else, selecting the memory structure whose address is stored as the selected memory element; and
  - (d) if the selected character is the last character of the query element:
    - determining if the first memory location of the memory structure selected in step (c) refers to the query element; and
      - if the first memory location of the memory structure selected in step (c) refers to the query element, outputting a signal indicating that the query element is included in the set of elements, else outputting a signal indicating that the query element is not included in the set of elements;
    - else:
      - selecting the next character of the query element , and repeating steps (c) and (d).

1                 11.     The method of claim 1 wherein a size of the data structure is  
2 independent of the number of elements in the set of elements.

1                 12.     The method of claim 11 wherein:  
2                     the set of elements contains elements from a domain  $\Sigma$  having a character set  
3                     of “m” characters, and wherein “Z” is the maximum possible length of an element in domain  
4                      $\Sigma$ ; and  
5                     the data structure comprises “Z” memory structures, each memory structure  
6                     comprising “m” slots, each slot comprising a first memory location and an array of memory  
7                     locations, each array of memory locations comprising “(m+1)” memory locations.

1                 13.     The method of claim 12 wherein building the data structure based upon  
2 information identifying the elements in the set of elements comprises:  
3                     initializing the first memory location and memory locations in the array of  
4                     memory locations of each slot in each memory structure to null values;  
5                     for each element “R” in the set of elements, where  $R = c_1c_2\dots c_f$  for some  
6                      $f \leq Z$ , for each  $c_i$  where  $1 \leq i \leq f$ :  
7                         if (“i” < “f”):  
8                             storing a non-null value in a memory location corresponding to  
9                     character  $c_{i+1}$  in the array of memory locations of the slot corresponding to  $c_i$  of memory  
10                     structure  $i$ ; and  
11                         if (“i” is equal to “f”):  
12                             storing a non-null value in the  $(m+1)^{th}$  memory location of the  
13                     array of memory locations of the slot corresponding to  $c_i$  of memory structure  $i$ ; and  
14                             storing a reference to element “R” in the first memory location  
15                     of the slot corresponding to  $c_i$  of memory structure  $i$ .

1                 14.     The method of claim 13 wherein:  
2                     receiving information identifying the query element comprises:  
3                         receiving information identifying a query element “k”, where  
4                      $k = c_1c_2\dots c_q$  for some  $q \leq Z$ ;  
5                     using the data structure to determine if the query element is included in the set  
6                     of elements comprises:

7 outputting a signal indicating that the query element is included in the  
8 set of elements if, for each  $c_i$  of  $k$ :

9 if (“ $i$ ” < “q”), a non-null value is stored in a memory location  
10 corresponding to character  $c_{i+1}$  in the array of memory locations of the slot corresponding to  
11  $c_i$  of memory structure  $i$ ; and

12 if (“ $i$ ” is equal to “ $q$ ”), a non-null value is stored in the  $(m+1)^{th}$   
13 memory location in the array of memory locations of the slot corresponding to  $c_i$  of memory  
14 structure  $i$ , and the first memory location of the slot corresponding to  $c_i$  of memory structure  $i$   
15 refers to the query element.

15. The method of claim 13 wherein:

2 receiving information identifying the query element comprises:

3 receiving information identifying a query element “k”, where

4       $k = c_1 c_2 \dots c_q$  for some  $q \leq Z$ ; and

5 using the data structure to determine if the query element is included in the set  
6 of elements comprises:

7 outputting a signal indicating that the query element is not included in  
 8 the set of elements if, for any  $c_i$  of  $k$ :

if ("*i*" is equal to "q"), a null value is stored in the  $(m+1)^{th}$

10 memory location in the array of memory locations of the slot corresponding to  $c_i$  of memory  
11 structure  $i$ , or the first memory location of the slot corresponding to  $c_i$  of memory structure  $i$   
12 does not refer to the query element; and

13                           if (“ $i$ ” < “q”), a null value is stored in a memory location  
14 corresponding to character  $c_{i+1}$  in the array of memory locations of the slot corresponding to  
15  $c_i$  of memory structure  $i$ .

1                   16. A system for determining if a query element is included in a set of  
2 elements, the system comprising:

3 a processor;

4 a memory coupled to the processor, the memory configured to store a plurality  
5 of code modules executable by the processor, the plurality of code modules comprising:

6 a code module for building a data structure based upon information  
7 identifying elements in the set of elements;

8                   a code module for receiving information identifying the query element;  
9       and

10                  a code module for using the data structure to determine if the query  
11       element is included in the set of elements such that the number of comparisons needed to  
12       determine if the query element is included in the set of elements is proportional to a length of  
13       the query element and independent of the number of elements in the set of elements.

1                  17.      The system of claim 16 wherein the query element is of length “q” and  
2       at most “q” character comparisons are needed to determine if the query element is included in  
3       the set of elements.

1                  18.      The system of claim 16 wherein:  
2                  the set of elements contains elements from a domain  $\Sigma$  having a character set  
3       of “m” characters, wherein “Z” is the maximum possible length of an element in domain  $\Sigma$   
4       and “Y” is the length of the longest element in the set of elements such that  $1 \leq Y \leq Z$ ; and  
5                  the code module for building the data structure comprises a code module for  
6       building the data structure comprising a plurality of memory structures headed by a root  
7       memory structure, each memory structure in the plurality of memory structures comprising a  
8       first memory location and an array of “m” memory locations.

1                  19.      The system of claim 18 wherein:  
2                  the data structure comprises a total of  $(Y+1)$  levels; and  
3                  each memory structure in the data structure belongs to a level  $L$ , where ( $0 \leq L$   
4        $\leq Y$ ), the level for a particular memory structure denoting the number of memory structures,  
5       starting with the root memory structure, that have to be traversed to reach the particular  
6       memory structure, the root memory structure belonging to level 0.

1                  20.     The system of claim 19 wherein the code module for building the data  
2       structure based upon information identifying the elements in the set of elements comprises:  
3                  for each element “R” in the set of elements, where  $R = c_1c_2\dots c_f$  for some  
4        $f \leq Y$ , for each  $c_i$  where  $1 \leq i \leq f$ , starting with  $i = 1$ :  
5                          (a) a code module for selecting a memory structure at level “ $(i-1)$ ”;  
6                          (b) if a memory location corresponding to character  $c_i$  in the array of  
7       memory locations of the presently selected memory structure does not refer to another

8 memory structure in the database, a code module for storing an address of a new memory  
9 structure at level “i” in the memory location corresponding to character  $c_i$  in the array of  
10 memory locations of the selected memory structure;  
11 (c) a code module for selecting the memory structure at level “i” whose  
12 address is stored in the memory location corresponding to character  $c_i$  in the array of memory  
13 locations of the presently selected memory structure;  
14 (d) if (“i” is equal to “f”), a code module for storing a reference to  
15 element “R” in the first memory location of the memory structure selected in step (c);  
16 (e) a code module for incrementing the value of “i” by one; and  
17 (f) a code module for repeating steps (b), (c), (d), and (e) for each  $c_i$   
18 where (“i”  $\leq$  “f”).

1 21. The system of claim 20 wherein:  
2 the code module for receiving information identifying the query element  
3 comprises:  
4 a code module for receiving information identifying a query element  $k$ ,  
5 where  $k = c_1c_2\dots c_q$  for some  $q \leq Z$  ;  
6 the code module for using the data structure to determine if the query element  
7 is included in the set of elements comprises:  
8 for each  $c_i$  of  $k$  where  $1 \leq i \leq f$ , starting with  $i = 1$ :  
9 (a) a code module for selecting a memory structure of the database at  
10 level “(i-1)”;  
11 (b) if a memory location corresponding to character  $c_i$  in the array of  
12 memory locations of the presently selected memory structure does not refer to another  
13 memory structure in the database, a code module for outputting a signal indicating that the  
14 query element is not included in the set of elements;  
15 (c) if the memory location corresponding to character  $c_i$  in the array of  
16 memory locations of the presently selected memory structure stores an address of a memory  
17 structure of the database at level “i”, a code module for selecting the memory structure at  
18 level “i” whose address is stored;  
19 (d) a code module for incrementing the value of “i” by one; and  
20 (e) a code module for repeating steps (b), (c), and (d) while (“i”  $\leq$  “q”)  
21 and the signal indicating that the query element is not included in the set of elements has not  
22 been output; and

23                   if the signal indicating that the query element is not included in the set of  
24 elements has not been output:  
25                    a code module for determining if the first memory location of the  
26 memory structure selected in step (c) refers to the query element; and  
27                    if the first memory location of the memory structure selected in step (c)  
28 refers to the query element, a code module for outputting a signal indicating that the query  
29 element is included in the set of elements, else a code module for outputting a signal  
30 indicating that the query element is not included in the set of elements.

1                   22. The system of claim 18 wherein the code module for building the data  
2 structure based upon information identifying the elements in the set of elements comprises:  
3                    for each element “R” in the set of elements, where  $R = c_1c_2\dots c_f$  for some  
4  $f \leq Y$ , where each character  $c_i$  belongs to the character set of domain  $\Sigma$ , and  $1 \leq i \leq f$ , a code  
5 module for storing information in the database indicating the position and identity of each  
6 character in element R.

1                   23. The system of claim 22 wherein the code module for using the data  
2 structure to determine if the query element is included in the set of elements comprises:  
3                    a code module for determining if the query element is included in the set of  
4 elements based upon information stored by the database and information identifying  
5 characters and their positions in the query element.

1                   24. The system of claim 18 wherein the code module for building the data  
2 structure based upon information identifying the elements in the set of elements comprises:  
3                    for each element “R” in the set of elements:  
4                      (a) a code module for selecting the root memory structure of the data  
5 structure as the selected memory structure;  
6                      (b) a code module for selecting the first character of element R;  
7                      (c) if a memory location corresponding to the selected character in the  
8 array of memory locations of the selected memory structure does not refer to another memory  
9 structure in the data structure, a code module for storing an address of a new memory  
10 structure in the memory location corresponding to the selected character in the array of  
11 memory locations of the presently selected memory structure;

25. The system of claim 24 wherein the code module for using the data structure to determine if the query element is included in the set of elements comprises:

- (a) a code module for selecting the root memory structure of the data structure as the selected memory structure;
- (b) a code module for selecting the first character of the query element;
- (c) if a memory location corresponding to the selected character in the array of memory locations of the selected memory structure does not refer to another memory structure in the data structure, a code module for outputting a signal indicating that the query element is not included in the set of elements,
  - else, a code module for selecting the memory structure whose address is stored as the selected memory element; and
  - (d) if the selected character is the last character of the query element:
    - a code module for determining if the first memory location of the memory structure selected in step (c) refers to the query element; and
      - if the first memory location of the memory structure selected in step (c) refers to the query element, a code module for outputting a signal indicating that the query element is included in the set of elements, else a code module for outputting a signal indicating that the query element is not included in the set of elements;
    - else:
      - a code module for selecting the next character of the query element , and repeating steps (c) and (d).

1                   26. The system of claim 16 wherein a size of the data structure is  
2 independent of the number of elements in the set of elements.

1 27. The system of claim 26 wherein:

2                   the set of elements contains elements from a domain  $\Sigma$  having a character set  
3   of "m" characters, and wherein "Z" is the maximum possible length of an element in domain  
4    $\Sigma$ ; and

5                   the data structure comprises "Z" memory structures, each memory structure  
6   comprising "m" slots, each slot comprising a first memory location and an array of memory  
7   locations, each array of memory locations comprising "(m+1)" memory locations.

1                 28.   The system of claim 27 wherein the code module for building the data  
2   structure based upon information identifying the elements in the set of elements comprises:

3                   a code module for initializing the first memory location and memory locations  
4   in the array of memory locations of each slot in each memory structure to null values;

5                   for each element "R" in the set of elements, where  $R = c_1c_2\dots c_f$  for some  
6    $f \leq Z$ , for each  $c_i$  where  $1 \leq i \leq f$ :

7                      if ("i" < "f"):

8                          a code module for storing a non-null value in a memory  
9   location corresponding to character  $c_{i+1}$  in the array of memory locations of the slot  
10   corresponding to  $c_i$  of memory structure  $i$ ; and

11                      if ("i" is equal to "f"):

12                          a code module for storing a non-null value in the  $(m+1)^{th}$   
13   memory location of the array of memory locations of the slot corresponding to  $c_i$  of memory  
14   structure  $i$ ; and

15                          a code module for storing a reference to element "R" in the first  
16   memory location of the slot corresponding to  $c_i$  of memory structure  $i$ .

1                 29.   The system of claim 28 wherein:

2                   the code module for receiving information identifying the query element  
3   comprises:

4                   a code module for receiving information identifying a query element  
5   " $k$ ", where  $k = c_1c_2\dots c_q$  for some  $q \leq Z$ ;

6                   the code module for using the data structure to determine if the query element  
7   is included in the set of elements comprises:

8                   a code module for outputting a signal indicating that the query element  
9   is included in the set of elements if, for each  $c_i$  of  $k$ :

10 if (“*i*” < “q”), a non-null value is stored in a memory location  
11 corresponding to character  $c_{i+1}$  in the array of memory locations of the slot corresponding to  
12  $c_i$  of memory structure *i*; and  
13 if (“*i*” is equal to “q”), a non-null value is stored in the  $(m+1)^{th}$   
14 memory location in the array of memory locations of the slot corresponding to  $c_i$  of memory  
15 structure *i*, and the first memory location of the slot corresponding to  $c_i$  of memory structure *i*  
16 refers to the query element.

1 30. The system of claim 28 wherein:  
2 the code module for receiving information identifying the query element  
3 comprises:

4 a code module for receiving information identifying a query element  
5 “k”, where  $k = c_1c_2\dots c_q$  for some  $q \leq Z$ ; and

6 the code module for using the data structure to determine if the query element  
7 is included in the set of elements comprises:

8 a code module for outputting a signal indicating that the query element  
9 is not included in the set of elements if, for any  $c_i$  of *k*:

10 if (“*i*” is equal to “q”), a null value is stored in the  $(m+1)^{th}$   
11 memory location in the array of memory locations of the slot corresponding to  $c_i$  of memory  
12 structure *i*, or the first memory location of the slot corresponding to  $c_i$  of memory structure *i*  
13 does not refer to the query element; and

14 if (“*i*” < “q”), a null value is stored in a memory location  
15 corresponding to character  $c_{i+1}$  in the array of memory locations of the slot corresponding to  
16  $c_i$  of memory structure *i*.

1 31. A computer program product stored on a computer-readable storage  
2 medium for determining if a query element is included in a set of elements, the computer  
3 program product comprising:  
4 code for building a data structure based upon information identifying elements  
5 in the set of elements;  
6 code for receiving information identifying the query element; and  
7 code for using the data structure to determine if the query element is included  
8 in the set of elements such that the number of comparisons needed to determine if the query

9 element is included in the set of elements is proportional to a length of the query element and  
10 independent of the number of elements in the set of elements.

1           32. The computer program product of claim 31 wherein the query element  
2 is of length “q” and at most “q” character comparisons are needed to determine if the query  
3 element is included in the set of elements.

1           33. The computer program product of claim 31 wherein:  
2                 the set of elements contains elements from a domain  $\Sigma$  having a character set  
3 of “m” characters, wherein “Z” is the maximum possible length of an element in domain  $\Sigma$   
4 and “Y” is the length of the longest element in the set of elements such that  $1 \leq Y \leq Z$ ; and  
5                 the code for building the data structure comprises code for building the data  
6 structure comprising a plurality of memory structures headed by a root memory structure,  
7 each memory structure in the plurality of memory structures comprising a first memory  
8 location and an array of “m” memory locations.

1           34. The computer program product of claim 33 wherein:  
2                 the data structure comprises a total of  $(Y+1)$  levels; and  
3                 each memory structure in the data structure belongs to a level  $L$ , where  $(0 \leq L$   
4  $\leq Y)$ , the level for a particular memory structure denoting the number of memory structures,  
5 starting with the root memory structure, that have to be traversed to reach the particular  
6 memory structure, the root memory structure belonging to level 0.

1           35. The computer program product of claim 33 wherein the code for  
2 building the data structure based upon information identifying the elements in the set of  
3 elements comprises:

4                 for each element “R” in the set of elements, where  $R = c_1c_2\dots c_f$  for some  
5  $f \leq Y$ , where each character  $c_i$  belongs to the character set of domain  $\Sigma$ , and  $1 \leq i \leq f$ , code  
6 for storing information in the database indicating the position and identity of each character  
7 in element R.

1           36. The computer program product of claim 35 wherein the code for using  
2 the data structure to determine if the query element is included in the set of elements  
3 comprises:

4 code for determining if the query element is included in the set of elements  
5 based upon information stored by the database and information identifying characters and  
6 their positions in the query element.

1 37. The computer program product of claim 31 wherein:  
2 a size of the data structure is independent of the number of elements in the set  
3 of elements;

4 the set of elements contains elements from a domain  $\Sigma$  having a character set  
5 of "m" characters, and wherein "Z" is the maximum possible length of an element in domain  
6  $\Sigma$ ; and

7 the data structure comprises "Z" memory structures, each memory structure  
8 comprising "m" slots, each slot comprising a first memory location and an array of memory  
9 locations, each array of memory locations comprising "(m+1)" memory locations.

1 38. The computer program product of claim 37 wherein the code for  
2 building the data structure based upon information identifying the elements in the set of  
3 elements comprises:

4 code for initializing the first memory location and memory locations in the  
5 array of memory locations of each slot in each memory structure to null values;

6 for each element "R" in the set of elements, where  $R = c_1c_2\dots c_f$  for some  
7  $f \leq Z$ , for each  $c_i$  where  $1 \leq i \leq f$ :

8 if ("i" < "f"):  
9 code for storing a non-null value in a memory location  
10 corresponding to character  $c_{i+1}$  in the array of memory locations of the slot corresponding to  
11  $c_i$  of memory structure  $i$ ; and

12 if ("i" is equal to "f"):  
13 code for storing a non-null value in the  $(m+1)^{th}$  memory  
14 location of the array of memory locations of the slot corresponding to  $c_i$  of memory structure  
15  $i$ ; and

16 code for storing a reference to element "R" in the first memory  
17 location of the slot corresponding to  $c_i$  of memory structure  $i$ .

1 39. The computer program product of claim 38 wherein:  
2 the code for receiving information identifying the query element comprises:

PCT/EP2019/052069

3 code for receiving information identifying a query element “k”, where  
4  $k = c_1c_2\dots c_q$  for some  $q \leq Z$ ;

5 the code for using the data structure to determine if the query element is  
6 included in the set of elements comprises:

7 code for outputting a signal indicating that the query element is  
8 included in the set of elements if, for each  $c_i$  of  $k$ :

9 if (“ $i$ ” < “ $q$ ”), a non-null value is stored in a memory location  
10 corresponding to character  $c_{i+1}$  in the array of memory locations of the slot corresponding to  
11  $c_i$  of memory structure  $i$ ; and

12 if (“ $i$ ” is equal to “ $q$ ”), a non-null value is stored in the  $(m+1)^{th}$   
13 memory location in the array of memory locations of the slot corresponding to  $c_i$  of memory  
14 structure  $i$ , and the first memory location of the slot corresponding to  $c_i$  of memory structure  $i$   
15 refers to the query element.

1 40. The computer program product of claim 38 wherein:

2 the code for receiving information identifying the query element comprises:

3 code for receiving information identifying a query element “k”, where

4  $k = c_1c_2\dots c_q$  for some  $q \leq Z$ ;

5 the code for using the data structure to determine if the query element is  
6 included in the set of elements comprises:

7 code for outputting a signal indicating that the query element is not  
8 included in the set of elements if, for any  $c_i$  of  $k$ :

9 if (“ $i$ ” is equal to “ $q$ ”), a null value is stored in the  $(m+1)^{th}$   
10 memory location in the array of memory locations of the slot corresponding to  $c_i$  of memory  
11 structure  $i$ , or the first memory location of the slot corresponding to  $c_i$  of memory structure  $i$   
12 does not refer to the query element; and

13 if (“ $i$ ” < “ $q$ ”), a null value is stored in a memory location  
14 corresponding to character  $c_{i+1}$  in the array of memory locations of the slot corresponding to  
15  $c_i$  of memory structure  $i$ .